WHAT IS CLAIMED IS:

1. A method for data synchronization, comprising:

generating a tree data structure containing data requiring synchronization between a web server and a client and a number of change flags that each correspond to a separate leaf node of the tree data structure and each indicating whether a revision has been made to leaf node data stored in the leaf node;

setting each of the number of change flags to a true value, if the corresponding leaf node data is revised;

transmitting the tree data structure, except the number of change flags, to the client, when an initial connection is requested by the client;

generating a number of first checksums, each for the leaf node data stored in a separate one of the number of leaf nodes stored by the web server;

comparing each of the number of first checksums to a corresponding one of a number of second checksums contained in a request datagram communicated to the web server by the client;

for each of the number of leaf nodes whose first checksum did not match its corresponding second checksum during the comparison, transmitting the leaf node data associated with the first checksum and a first event number, associated with the leaf node data, to the client in a response datagram;

inspecting the value of each of the number of the change flags occasionally; and

for each of the number of change flags set to the true value, transmitting the corresponding leaf node data in an event datagram to the client.

2. The method of claim 1, wherein each of the number of leaf nodes comprises:

a data location identifier (DataLocID) indicating a location of the leaf node;

the change flag (ChangeFlag) corresponding to the leaf node;

a DataSize identifying a size of the leaf node data; and

the leaf node data regarding a communication device.

3. The method of claim 1, further comprising:

observing an arrival, from a communication device, of an event containing an information change pertaining to the communication device;

revising the leaf node data of the leaf node corresponding to the communication device to reflect the information change; and

setting the change flag of the leaf node revised by the information change of the event to a true value.

4. The method of claim 1, wherein the request datagram includes a DatagramSize that identifies an entire size of the datagram, a DatagramType that identifies a type of the datagram, and the second number of checksums.

5. The method of claim 1, wherein the response datagram includes a DatagramSize that identifies an entire size of the datagram, a DatagramType that identifies a type of the datagram, an EventNum representing a loss of the event datagram, and the leaf node data.

6. The method of claim 1, wherein the event datagram includes a DatagramSize identifying an entire size of the datagram, a DatagramType that identifies a type of the datagram, an EventNum representing the loss of the event datagram, and the leaf node data.

7. The method of claim 1, wherein the change flag is set in the leaf node corresponding to a communication device in accordance with a report of an information change in the corresponding communication device, and is reset by the web server when the information change is transmitted to the client.

8. A method for data synchronization, comprising:

receiving a tree data structure requiring synchronization between a web server and a client by connecting to the web server;

generating a checksum of leaf node data in each of a number of leaf nodes in the received tree data structure and initializing a first event number for judging a loss of data;

comparing the first event number to a second event number included in an event datagram, when the event datagram is transmitted from the web server;

increasing the first event number by a certain degree, updating the leaf node data of a first set of leaf nodes of the number of leaf nodes in the tree data structure using data included in the event datagram, and regenerating the checksums of the updated first set of leaf nodes, if the first event number and the second event number are the same;

transmitting the checksums of every leaf node of the number of leaf nodes in the tree data structure in one request datagram to the web server, if the first event number and the second event number are different from each other; and

updating the leaf node data of a second set of leaf nodes of the number of leaf nodes in the tree data structure using the data included in a response datagram, updating the first event number, and regenerating the checksums of the updated second set leaf nodes, when the response datagram is transmitted from the web server.

9. The method of claim 8 further comprising:

transmitting periodically the checksums of every leaf node of the number of leaf nodes in the tree data structure of the client in the request datagram to the web server;

updating the leaf node data of a third set of leaf nodes of the number of leaf nodes and the first event number in the tree data structure of the client using the response datagram, when the response datagram is transmitted from the web server; and

regenerating the checksums of the updated third set of leaf nodes.

10. The method of claim 8, wherein the event datagram and the response datagram each include a DatagramSize that identifies an entire size of the corresponding datagram, a DatagramType identifying a type of the corresponding datagram, an EventNum representing whether or not the event datagram is lost, and the data.

11. The method of claim 10, wherein the event datagram and the response datagram are distinguished by the value of the DatagramType.

12. The method of claim 8, wherein the request datagram includes a DatagramSize that identifies an entire size of the datagram, a DatagramType that identifies a type of the datagram, and the checksums of every leaf node of the number of leaf nodes.

13. A method for data synchronization, comprising:

generating a first tree data structure requiring synchronization between a web server and a client in a web-based communication device management system and transmitting the first tree data structure to the client as a second tree data structure when the client initially connects to the web server;

inspecting a change flag of each leaf node of a number of leaf nodes in the first tree data structure periodically to detect whether a change of leaf node data has occurred within any leaf node;

transmitting the leaf node data of every leaf node in the first tree data structure, having a change of leaf node data, in an event datagram at each period to the client;

inspecting whether or not an event is lost by comparing a first event number of the client to a second event number included in the event datagram, when the event datagram is received by the client;

transmitting a request datagram comprising a second checksum for each leaf node of the number of leaf nodes of the second tree data structure to the web server, if the event is lost;

generating a first checksum of each leaf node of the number of leaf nodes in the first tree data structure, and judging whether a change has occurred to the first tree data structure by comparing the corresponding first checksum and second checksum of each leaf node in the first and second tree data structures, respectively, when the request datagram is transmitted to the web server;

transmitting a response datagram comprising both the leaf node data of each of the number of leaf nodes in the first tree data structure having an associated first checksum that differs from the corresponding second checksum and the second event number to the client, if at least one first checksum and its corresponding second checksum differ;

updating the leaf node data of each of the number of leaf nodes in the second tree data structure having an associated first checksum that differs from the corresponding second checksum using the corresponding leaf node data of the first tree data structure in the response datagram; and

updating the leaf node data of each of the number of leaf nodes in the second tree data structure having a corresponding first tree data structure leaf node in the event datagram, if the event is not lost.

14. The method of claim 13, wherein the change flag of the corresponding leaf node is set according to a report of changing corresponding management data by a communication device, and is reset by the web server when the management data is communicated to the client.

15. The method according of 13, wherein the second event number is increased by a certain degree whenever the event datagram is transmitted, and the first event number is increased by the certain degree only when the event datagram is received without any loss.

16. The method of claim 13, wherein the client periodically transmits the request datagram.

17. A method for data synchronization, comprising:

for each of a number of communication devices, generating a server leaf node within a server tree data structure comprising management data of the respective communication device and a change flag;

communicating the server tree data structure to a client of a client server system as a client tree data structure, wherein each server leaf node has a corresponding client leaf node within the client tree structure;

setting the change flag of the corresponding server leaf node when the management data of one of the number of communication devices is revised; and

resetting the change flag of the corresponding server leaf node when the revised management data is communicated by the server to the client.

18. The method of claim 17, further comprising:

occasionally determining whether the change flag of any of the number server leaf nodes is set; and

communicating an event datagram from the server to the client comprising each of the number of server leaf nodes, having a set change flag, and a server event number.

19. The method of claim 17, further comprising generating a checksum of the management data in each of the number of client leaf nodes and storing each of the number of checksums as a client checksum set.

20. The method of claim 19, further comprising:

occasionally communicating the client checksum set to the server in a request datagram;

generating a checksum of the management data in each of the number of server leaf nodes; and

communicating each of the server leaf nodes, of the number of server leaf nodes, having a respective checksum that differs from the corresponding checksum of the client checksum set, and a server event number to the client in a response datagram.

21. The method of claim 20, further comprising:

(a) revising the management data of each client leaf node of the client tree data structure having a corresponding server leaf node within the response datagram, using the management data of the respective server leaf node; and

(b) regenerating the checksum of the management data in each of the number of client leaf nodes revised in step (a).

22. The method of claim 18, further comprising:

(a) revising the management data of each client leaf node of the client tree data structure having a corresponding server leaf node within the event datagram, using the management data of the respective server leaf node; and

(b) regenerating the checksum of the management data in each of the number of client leaf nodes revised in step (a);

(c) changing a client event number by a predetermined amount;

(d) communicating the client checksum set to the server in a request datagram,

wherein, steps (a), (b), and (c) are performed if the server event number contained in the event datagram has a value corresponding to the client event number and step (d) is performed if the server event number contained in the event datagram has a value that does not correspond to the client event number.

23. The method of claim 17, further comprising, for each of the number of communication devices, communicating the management data regarding the respective communication device to the server when the management data is revised.

24. The method of claim 20, wherein the event datagram and response datagram each contain a datagram type and are distinguished by their respective datagram type values.

25. A data structure for data synchronization between a web server and a client comprising:

a parent node that has a parent identification (ID) field and a last leaf node location ID field; and

a plurality of leaf nodes that are associated with the parent node, each leaf node having a location ID field, a change flag field, a data size field, and a data field.

26. The data structure of claim 25, wherein the change flag field indicates whether a revision of information within the respective leaf node has occurred subsequent to a prior event.

27. The data structure of claim 25, wherein the change flag field is set when a revision of information within the respective leaf node has occurred subsequent to a prior event and is reset when the data synchronization has occurred between the web server and the client.

28. The data structure of claim 25, wherein;

the last leaf node location ID identifies a location of a last of the plurality of leaf nodes;

the location ID field identifies a position of the respective leaf node;

the change flag field indicates whether a revision of information within the respective leaf node has occurred subsequent to a prior event;

the data size field identifies a size of management data corresponding to the respective leaf node; and

the data field contains the management data of the respective leaf node.

29. A data structure for data synchronization between a web server and a client comprising:

a datagram size field that identifies an entire size of the data structure;

a datagram type field that identifies a type of the data structure;

an event number field that identifies a synchronization event between the web server and the client; and

a management data field that contains revised management data.

30. The data structure of claim 29, wherein;

the client updates client management data based on the revised management data communicated by the server; and

the client determines whether the data synchronization has been lost based on a value of the event number.